



22883

PATENT TRADEMARK OFFICE

103.1071.01

This application is submitted in the name of the following inventors:

<i>Inventor</i>	<i>Citizenship</i>	<i>Residence City and State</i>
Robert ENGLISH	United States	Menlo Park, California
Szu-Wen KUO	Taiwan	Milpitas, California
Brian QUIRION	United States	San Jose, California

The assignee is *Network Appliance, Inc.*, a California corporation having an office at 495 East Java Drive, Sunnyvale, CA 94089.

## TITLE OF THE INVENTION

Managing Configuration Information for Multiple Devices

## BACKGROUND OF THE INVENTION

### 1. *Field of the Invention*

This invention relates to managing configuration information for multiple devices.

2. *Related Art*

Each computer system includes information about its internal state, such as where its software elements are located, where data is maintained for those software elements, and the like. As computer systems become relatively large, the amount of that information also becomes relatively large, and the task maintaining that information in a manageable form becomes relatively difficult. For one example, in a cache management system (a system including a set of caching devices each of which maintains relatively local copies of web pages and possibly other network objects obtained from relatively remote origination servers), each one of a set of multiple caches can include information regarding how web pages should be downloaded, how quickly, whether selected client devices can download web pages, where selected web pages would be downloaded from, and the like; when there are many such caches, assuring that each cache has correct and up-to-date information can be detailed and time-consuming. For a second example, there can be multiple devices each having at least some information that is unique for each device and at least some information that is identical for multiple devices. In both examples, management of such information, including assuring that each device has correct and up-to-date information, can be detailed and time-consuming, and therefore error-prone.

A first problem in the known art is that access to multiple devices involves either direct physical access or communication with each such device. Direct physical

connections to multiple devices can be inconvenient, such as in those cases where the set of devices is physically dispersed. Using a communication link is subject to the drawback that at least some of the devices might be unavailable at the time. Moreover, both of these methods presume that the manager already knows of the existence (and how to reach) each of the multiple devices, to direct them to use the particular information the manager selects.

Additional problems in the known art include the following:

- There is no convenient method in the known art for managing both local and remote information for multiple such devices.
- There is no convenient method in the known art for managing multiple such devices in groups or subgroups.
- There is no convenient method in the known art for managing device information by multiple information managers.
- There is no convenient method in the known art for managing device information when those devices are protected by firewalls, without granting access to those devices to managers.

1           Accordingly, it would be advantageous to provide an improved technique  
2 for managing information, particularly configuration information, for multiple devices.

## 3 4                           SUMMARY OF THE INVENTION

5  
6           The invention provides a method and system for managing device informa-  
7 tion for multiple devices, particularly configuration information. This is achieved in an  
8 embodiment of the invention in which each one of a set of devices retrieves information  
9 from a sequence of information servers (preferably both local and remote).

10  
11           In a preferred embodiment, a device when starting up reads a list of con-  
12 figuration file names; reads configuration information from each configuration file in se-  
13 quence; and resolves conflicts among configuration files to present a consistent configu-  
14 ration on each restart of each device.

15  
16           In a preferred embodiment, a configuration file includes a sequence of  
17 <name, value> pairs, each of which represents a variable having the name <name> and  
18 the value <value>. In those cases when two variables have the same name, an operator  
19 associated with the second <name, value> pair indicates whether to overwrite the first  
20 value or to edit the first value, such as by appending the second value.

1 In a preferred embodiment, one of the <name, value> pairs indicates the list  
2 of configuration file names itself. When the list of configuration file names is changed,  
3 the device for which the change is made re-reads the sequence of configuration files and  
4 repeats its determination of the <name, value> pairs, until the list of configuration file  
5 names is stabilized (until it is possible to re-read the sequence of configuration files with-  
6 out further changes).

7  
8 In a preferred embodiment, a device when starting up reads a value for the  
9 list of configuration file names, equal to the stabilized value determined by the device at  
10 the most recent restart.

11  
12 Those of ordinary skill in the art will recognize, after perusal of this appli-  
13 cation, the many advantages provided by the invention. These include, but are not limited  
14 to, the following:

- 15
- 16 • maintaining information other than configuration information, such as for example  
17 information to be communicated among devices or users;
  - 18
  - 19 • maintaining information by multiple managers or by a hierarchy of managers, such  
20 as for example when each manager is assigned a subgroup of the information or a  
21 subgroup of the devices to manage;
  - 22

- maintaining information in multiple locations, such as for example when at least part of the information can be found local to devices and at least part of the information can be found remotely from devices.

The invention has general applicability to maintaining information for multiple devices. Although a preferred embodiment is described with regard to configuration information for a set of multiple caches, there is no particular limitation of the invention to caches or similar devices, nor is there any particular limitation of the invention to configuration information. Techniques used by a preferred embodiment of the invention for remotely maintaining information for multiple devices, for resolving conflicts at devices between differing sets of information, and for remotely maintaining information by multiple managers, can be used in contexts other than the specific applications disclosed herein.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a block diagram of a system capable of managing information for a set of devices.

Figure 2 shows a process flow diagram of a method including steps of managing information for a set of devices.

## 1 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

2  
3 In the following description, a preferred embodiment of the invention is de-  
4 scribed with regard to preferred process steps and data structures. Those skilled in the art  
5 would recognize after perusal of this application that embodiments of the invention can  
6 be implemented using one or more general purpose processors or special purpose proces-  
7 sors or other circuits adapted to particular process steps and data structures described  
8 herein, and that implementation of the process steps and data structures described herein  
9 would not require undue experimentation or further invention.

10  
11 *Lexicography*

12  
13 The following terms refer or relate to aspects of the invention as described  
14 below. The descriptions of general meanings of these terms are not intended to be limit-  
15 ing, only illustrative.

- 16  
17 • **information to be maintained** — in general, any values or other data to be re-  
18 corded in selected locations for possible retrieval by devices for which that infor-  
19 mation is maintained.

- 1     • **maintaining information** — in general, any operation for creating, modifying,  
2       deleting, assigning, or otherwise managing information for possible use by de-  
3       vices.
- 4
- 5     • **local and remote information** — in general, local information is that which is ac-  
6       cessible by a device without cooperation with any other device, while remote in-  
7       formation is that which is accessible by a device using a communication link or a  
8       communication protocol to record or retrieve that information.
- 9
- 10    • **device groups** — in general, any technique for associating sets of devices with an  
11      identifier selecting those devices. Device groups can have zero, one, or more  
12      members, and can overlap.
- 13
- 14    • **configuration information** — in general, any information used during an initial  
15      (“start-up”) phase of operation for a device. There is no particular limitation of the  
16      invention to information that is not used after start-up.
- 17
- 18    • **configuration file names** — in general, any identifier allowing a device to identify  
19      a source for configuration information. For one example, such “file names” might  
20      be URLs or database queries, rather than names in a filesystem namespace.
- 21



As noted above, these descriptions of general meanings of these terms are not intended to be limiting, only illustrative. Other and further applications of the invention, including extensions of these terms and concepts, would be clear to those of ordinary skill in the art after perusing this application. These other and further applications are part of the scope and spirit of the invention, and would be clear to those of ordinary skill in the art, without further invention or undue experimentation.

### *System Elements*

Figure 1 shows a block diagram of a system capable of managing information for a set of devices.

A system 100 includes a set of devices 110, a set of configuration servers 120, a set of management consoles 130, and a communication link 140. These elements operate together as described to perform the functions shown herein for the system 100.

### Devices

Each device 110 preferably includes a processor, program and data memory, mass storage, and a network interface to the communication link 140. Each device 110 is therefore capable of performing programmed processes and communicating with other elements of the system 100 using the communication link 140. The program and

1 data memory in each device 110 include computer programs and data structures for per-  
2 forming steps described in this application. There is no particular requirement that the  
3 devices 110 must include mass storage, merely that at least some of their storage is per-  
4 sistent across restart operations. References herein to mass storage for the devices 110  
5 are intended to be read broadly to include other forms of persistent storage and techniques  
6 for retaining information for restarting and communicating with other elements of the  
7 system 100.

8  
9 Each device 110 includes a set of local configuration files 111, having in-  
10 formation for use by the device 110. In a preferred embodiment, these local configuration  
11 files 111 include at least a default configuration file 111a and a last-boot configuration  
12 file 111b. The default configuration file 111a includes information for the device 110  
13 determined by a manufacturer for the device 110. The last-boot configuration file 111b  
14 includes information for the device 110 as determined by the device 110 at of its last boot  
15 time, as described further in this application.

#### 16 17 Configuration Servers

18  
19 Similar to the devices 110, each configuration server 120 preferably in-  
20 cludes a processor, program and data memory, mass storage, and a network interface to  
21 the communication link 140. Each configuration server 120 is therefore capable of per-  
22 forming programmed processes and communicating with other elements of the system

1 100 using the communication link 140. Similar to the devices 110, the program and data  
2 memory in each configuration server 120 include computer programs and data structures  
3 for performing steps described in this application. There is no particular requirement that  
4 the configuration servers 120 must include mass storage, merely that at least some of their  
5 information is retained sufficient for restarting and communicating with other elements of  
6 the system 100.

7  
8 One or more (or all of the) configuration servers 120 can be located on the  
9 same physical device as one of the devices 110, in which case those devices 110 also in-  
10 clude program and data information for performing the functions described herein for a  
11 configuration server 120.

12  
13 Each configuration server 120 includes a set of remote configuration files  
14 121, having information for use by one or more devices 110. There is no particular re-  
15 quirement for any arrangement or hierarchy of remote configuration files 121. However,  
16 in a preferred embodiment, the remote configuration files 121 include at least one global  
17 configuration file 121a, and a set of group configuration files 121b. The global configu-  
18 ration file 121a includes information determined by at least one manager 131 (as further  
19 described below) to be for use by substantially all devices 110. The group configuration  
20 files 121b include information determined by at least one manager 131 (as further de-  
21 scribed below) to be for use by devices 110 associated with one or more device groups  
22 122.

1  
2 For example, a first device group 122 might be associated with those de-  
3 vices in an engineering department of a company, while a second device group 122 might  
4 be associated with those devices in a marketing department of a company. There is no  
5 particular requirement regarding the arrangement or hierarchy of device groups 122. For  
6 example, some devices 110 might be associated with more than one device group 122  
7 (such as for example an “engineering” device group 122 and an “executive” device group  
8 122).

9  
10 The remote configuration files 121 are accessible by the devices 110 using  
11 one or more information transfer protocols. Accordingly, each device 110 includes com-  
12 puter programs and data structures capable of preparing request messages and responding  
13 to response messages in at least one such information transfer protocol. Similarly, each  
14 configuration server 120 includes computer programs and data structures capable of re-  
15 sponding to request messages and preparing response messages in at least one such in-  
16 formation transfer protocol. There is no particular requirement that the information trans-  
17 fer protocol be any specific protocol. In a preferred embodiment, the information transfer  
18 protocols include HTTP, SHTTP, some similar protocol, or a combination or variant  
19 thereof. However, in alternative embodiments, the information transfer protocols may in-  
20 clude FTP, FSP, MUD, RSH, an embodiment-specific protocol using a selected port, or a  
21 combination or variant thereof.

1           Although there is no particular requirement therefore, in a preferred em-  
2 bodiment one or more devices 110 might also include remote configuration files 121  
3 (“remote” in the sense that they are remote to other devices 110), accessible by the de-  
4 vices 110 using one or more information transfer protocols. Accordingly, each such de-  
5 vice 110 also includes computer programs and data structures capable of responding to  
6 request messages and preparing response messages in at least one such information trans-  
7 fer protocol.

8  
9           In a preferred embodiment, each device 110 includes a cache, and is re-  
10 sponsive to HTTP, SHTTP, and similar protocols. Each device is also responsive to local  
11 file access (by the device 110 itself), remote file access (both other devices 110), FTP,  
12 RSH, and similar protocols.

#### 13 14 Management Consoles

15  
16           Similar to the devices 110 and the configuration servers 120, each manage-  
17 ment console 130 preferably includes a processor, program and data memory, mass stor-  
18 age, and a network interface to the communication link 140. Each management console  
19 130 is therefore capable of performing programmed processes and communicating with  
20 other elements of the system 100 using the communication link 140. Similar to the de-  
21 vices 110 and the configuration servers 120, the program and data memory in each con-  
22 figuration server 120 includes computer programs and data structures for performing

1 steps described in this application. There is no particular requirement that the manage-  
2 ment console 130 must include mass storage.

3  
4 Similar to the devices 110 and the configuration servers 120, one or more  
5 (or all of the) management consoles 130 can be located on the same physical device as  
6 one of the devices 110, in which case those devices 110 also include program and data in-  
7 formation for performing the functions described herein for a configuration server 120.

8  
9 Each management console 130 has associated with it a manager 131 (as  
10 further described above), such as a human operator capable of interacting with the man-  
11 agement console 130 to create, edit, and delete both local configuration files 111 and re-  
12 mote configuration files 121, and to assign them to devices 110. There is no particular  
13 requirement that the manager 131 is actually a human operator; in alternative embodi-  
14 ments, the manager 131 can include a program for automatic management of configura-  
15 tion information or artificial intelligence software for performing general-purpose tasks.

#### 16 17 Communication Link

18  
19 In a preferred embodiment, the communication link 140 includes a LAN  
20 (local area network). However, in alternative embodiments, the communication link 140  
21 may include any technique for communicating information, including without limitation a

1 WAN (wide area network), an enterprise network, a VPN (virtual private network), a pri-  
2 vate or public switched network, or some combination or variant thereof.

3  
4 There is no particular requirement that the communication link 140 must  
5 have any particular physical form, only that devices 110, configuration servers 120, and  
6 management consoles 130 can communicate with each other using the communication  
7 link 140. In a preferred embodiment, each element of the system 100 is accessible by  
8 each other element using a protocol such as HTTP, SHTTP, or a variant thereof. How-  
9 ever, there is no particular requirement that such HTTP access must be direct access; the  
10 communication link 140 might include intermediate devices such as proxies, firewalls,  
11 NAT translators, and the like.

### 12 13 *Method of Operation*

14  
15 Figure 2 shows a process flow diagram of a method including steps of man-  
16 aging information for a set of devices.

17  
18 A method 200 includes a set of flow points and process steps as described  
19 herein.

20  
21 Although by the nature of textual description, the flow points and process  
22 steps are described sequentially, there is no particular requirement that the flow points or

1 process steps must be sequential. Rather, in various embodiments of the invention, the  
2 described flow points and process steps can be performed in a parallel or pipelined man-  
3 ner, either by one device performing multitasking or multithreading, or by a plurality of  
4 devices operating in a cooperative manner. Parallel and pipelined operations are known  
5 in the art of computer science.

6  
7 At a flow point 210, the system 100 is ready to set configuration informa-  
8 tion for a device 110.

9  
10 At a step 211, the device 110 restarts. Any individual device restart can be  
11 in response to one or more of a set of factors, including without limitation a hardware in-  
12 terrupt, a software panic (sometimes known as a “crash”), an operator command, a sched-  
13 uled operation or some combination or variant thereof.

14  
15 At a step 212, the device 110 reads its last-restart configuration file 111b.  
16 As part of this step, the device 110 reads a sequence of <name, value> pairs and sets the  
17 value of a set of variables, one for each <name> (having that name) equal to the corre-  
18 sponding <value>. One of these variables is a FILE-LIST variable, which describes the  
19 sequence of locations in which the device 110 looks for configuration information.

20  
21 In a preferred embodiment, the FILE-LIST variable has a syntax including  
22 an initial delimiter (shown below as “\\”), a list of URL (uniform resource locator) val-



ues each on a separate line, and a final delimiter (shown below as “\\”). There is no particular requirement for using this syntax or these particular delimiters; any syntax or delimiters that could be consistently parsed by the devices 110 would suffice. Note that one of the URLs recited in the FILE-LIST variable has an “ftp:” prefix, indicating that the file is to be obtained using the FTP protocol.

```
\\
default.cfg
ftp://ncl-sun-0.lab.netapp.com/pub/global.cfg
local.cfg
\\
```

In a preferred embodiment, the default configuration file 111a (shown above as “default.cfg”) is always named first in the list. A preferred embodiment explicitly recites the location of the default configuration file 111a in the FILE-LIST variable (which will be a variable named in the last-restart configuration file 111b). However, in alternative embodiments, the default configuration file 111a may be implicitly recited, that is, not recited but simply understood by the device 110 to come first.

At a step 213, the device 110 determines the value of the FILE-LIST variable.

At a step 214, the device 110 reads each file named in the FILE-LIST variable in sequence. As part of this step, the device 110 performs the following sub-steps:

At a sub-step 214(a), the device 110 selects a next file from the FILE-LIST variable. Initially, the “next” file is the first file designated by the FILE-LIST variable.

Although these sub-steps 214(a), 214(b), 214(c), and 214(d) are described as if changes to variables are made immediately, in a preferred embodiment, the device 110 assures that all files specified by the FILE-LIST variable are accessible before making those changes. Each device 100 thus avoids using a specified configuration until complete copies of all configuration files have been obtained, so that it can assure that when the values in each configuration file are applied, they include a complete set.

At a sub-step 214(b), the device 110 determines if selected file has been updated. If not, the device 110 skips that file and continues with the sub-step 214(d). Otherwise, the device 110 continues with the next sub-step.

At a sub-step 214(c), the device 110 reads each <name, value> pair from the selected file in turn. In a preferred embodiment, each <name, value> pair appears in one of the following syntactic forms:

1

- 2 • name = value

3

- 4 • name += value

5

6 In either case, if the variable named <name> is not yet defined, it is defined  
7 and set equal to the associated value <value>.

8

9 If the variable named <name> is already defined, there is a difference in  
10 treatment depending on the choice of form. For the first syntactic form, the  
11 variable is set equal to the new value <value> without regard for its earlier  
12 value. In the second syntactic form, the variable has the new value ap-  
13 pended to the earlier value.

14

15 At a sub-step 214(d), the device 110 determines if there are any further files  
16 to read. If so, the device 110 loops back to the sub-step 214(a). Otherwise,  
17 the device 110 continues with the next step.

18

19 At a step 215, the device 110 saves its set of variables in the last-restart  
20 configuration file 111b, thus preserving the values associated with those variables for the  
21 next time the device 110 is restarted.

At a step 216, the device 110 determines if the FILE-LIST variable has been changed by performance of the step 214. If so, the method 200 goes back to the step 214. Accordingly, the method 200 will perform the step 214, the step 215, and the step 216 repeatedly until the FILE-LIST variable reaches a settled value.

At a flow point 220, the device 110 has set all its variables using the information in the local configuration files 111 and remote configuration files 121, and is ready to run.

### *Generality of the Invention*

The invention has general applicability to maintaining information for multiple devices. Although a preferred embodiment is described with regard to configuration information for a set of multiple caches, there is no particular limitation of the invention to caches or similar devices, nor is there any particular limitation of the invention to configuration information. Techniques used by a preferred embodiment of the invention for remotely maintaining information for multiple devices, for resolving conflicts at devices between differing sets of information, and for remotely maintaining information by multiple managers, can be used in contexts other than the specific applications disclosed herein.

1           Other and further applications of the invention in its most general form  
2 would be clear to those skilled in the art after perusal of this application. The invention  
3 would be usable for such other and further applications without undue experimentation or  
4 further invention.

5  
6           Although preferred embodiments are disclosed herein, many variations are  
7 possible which remain within the concept, scope and spirit of the invention; these varia-  
8 tions would be clear to those skilled in the art after perusal of this application.

TO: EL 734816049 US